Jurnal Ilmu Komputer Agri-Informatika

Tersedia secara *online* di:
http://journal.ipb.ac.id/index.php/jika

Volume 5 Nomor 1 halaman 1 – 10
ISSN: 2089-6026

Metode Klasifikasi Rocchio untuk Analisis Hoax

Rocchio Classification Method for Hoax Analysis

AULIA AFRIZA[1*], JULIO ADISANTOSO[1]

Abstrak

Hoax adalah informasi sesat dan berbahaya karena menyesatkan persepsi manusia dengan menyampaikan informasi palsu sebagai kebenaran. *Hoax* sendiri dapat bertujuan untuk mempengaruhi pembaca dengan informasi palsu sehingga pembaca mengambil tindakan sesuai dengan isi *hoax*. Untuk mengetahui informasi yang tersebar, maka diperlukan klasifikasi untuk mengetahui apakah informasi tersebut *hoax* atau bukan. Klasifikasi yang akan digunakan dalam penelitian ini adalah klasifikasi Rocchio, dimana hasil klasifikasi Rocchio akan dibandingkan dengan Multinomial Naive Bayes. Evaluasi pada penelitian ini menggunakan *confusion matrix,* dimana akurasi Rocchio didapatkan sebesar 83.501% sedangkan Multinomial Naive Bayes sebesar 65.835%.

Kata Kunci**:** *Hoax*, *Non-Hoax*, Rocchio, Multinomial Naive Bayes, *confusion matrix*.

Abstract

Hoax was misguided and dangerous information because misleading perception of humans by passing false information as truth. Hoax themselves can aim to influence the readers with false information so that readers take action in accordance with the content of the hoax. To find out the information that was spread, then required classification to determine whether such information was a hoax or not. The classification to be used in this research was the classification of Rocchio, where classification results would be compared Rocchio with Multinomial Naive Bayes. Evaluation on research used a confusion matrix, where accuracy was obtained by Rocchio of 83,501 whereas Multinomial Naive Bayes of 65,835%.

Keywords: Hoax, Non-Hoax, Rocchio, Multinomial Naive Bayes, confusion matrix.

PENDAHULUAN

Hoax adalah informasi sesat dan berbahaya karena menyesatkan persepsi manusia dengan menyampaikan informasi palsu sebagai kebenaran. *Hoax* mampu mempengaruhi banyak orang dengan menodai suatu citra dan kredibilitas (Rasywir dan Purwarianti 2015). *Hoax* sendiri dapat bertujuan untuk mempengaruhi pembaca dengan informasi palsu sehingga pembaca mengambil tindakan sesuai dengan isi *hoax*. Menurut Anna (2017), keterbatasan rentang perhatian manusia ditambah dengan informasi yang sangat deras di media sosial dianggap menjadi penyebab *hoax* gampang menyebar. Penelitian menyebutkan, seseorang cenderung melihat "bias informasi" dan hanya menaruh perhatian, serta menyebarkan informasi yang sesuai dengan kepercayaannya. Bahkan meski infomasi tersebut palsu.

Perkembangan tekhnologi informasi turut serta mendorong penyebaran berita atau informasi *hoax*. Menurut Pakpahan (2017), Teknologi Informasi untuk Indonesia sendiri ikut berkembang pesat dimana pengguna internet di Indonesia saat ini berjumlah 132,7 juta atau 52% dari jumlah penduduk Indonesia. Untuk mengetahui berita yang tersebar, maka diperlukan klasifikasi dokumen untuk mengetahui apakah berita tersebut *hoax* atau bukan. Klasifikasi dokumen merupakan proses menemukan sekumpulan model yang mendeskripsikan dan membedakan kelas-kelas data sesuai dengan kategori yang dimilikinya. Tujuan klasifikasi untuk memprediksikan kelas dari objek yang belum diketahui kelasnya dengan karakteristik tipe data yang bersifat kategorik (Han *et al.* 2011).

[1]Departemen Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Pertanian Bogor, Bogor 16680
*Penulis Korespondensi: Surel: auliaafriza@gmail.com

Penelitian terkait *hoax* pernah dilakukan oleh Petkovic et al, 2005, Vukovic et al, 2009, Chen et al.2014 dan Rasywir dan Purwarianti 2015, namun penelitian tersebut terkait domain email *hoax* dan eksperimen system klasifikasi untuk berita *hoax* yang menggunakan metode Levenshtein Distance, Fuzzy Logic, Feed Forward Neural Network, Naïve Bayes, Support Vector Machine dan Algoritma C4.5. Kesimpulan pada penelitian analisis sentimen sebelumnya dilakukan oleh Pantouw (2017) mengenai klasifikasi Rocchio dan Multinomial Naive Bayes pada pesan Twitter menunjukkan akurasi yang cukup baik yaitu 85.399% untuk model Multinomial dan 96.283% untuk klasifikasi Rocchio.

Penelitian yang menelaah tentang data berita *hoax* berbahasa indonesia maupun bahasa inggris dengan klasifikasi Rocchio masih belum dilakukan, sedangkan menurut Manning *et al. (*2009) klasifikasi Rocchio memiliki cara terbaik dalam menghitung batas kelas yang mana menggunakan *centroid* untuk menentukan batas-batasnya dan klasifikasi Rocchio dipilih karena memiliki kinerja yang lebih baik dibandingkan Multinomial Naive Bayes (Joachims ,1997). Pada penelitian Joachims (1997) dilakukan penelitian menggunakan kalsifikasi Rocchio namun pada data 1000 artikel dari 20 *newsgroup.* Oleh karena itu, penelitian ini bertujuan untuk mengklasifikasikan data berita *hoax* berbahasa Indonesia dengan membandingkan akurasi antara Rocchio dengan Multinomial Naïve Bayes berdasarkan penelitian Joachims (1997) dan sumber data yang digunakan didapatkan dari *broadcast messages, social media,* dan beberapa sumber media berita *online* seperti CNN.

METODE

Penelitian ini terdiri dari 6 tahap yaitu pengumpulan dokumen, praproses, seleksi fitur, pembagian data, klasifikasi dan evaluasi.

Pengumpulan Dokumen

Pengumpulan dokumen dilakukan dengan mengumpulkan 300 dokumen *hoax* dan 300 dokumen *non-hoax* berbahasa indonesia yang didapat dari *website, broadcast messages*, dan media social yang dikumpulkan pada Juli 2017 hingga 15 Oktober 2017. Dokumen tersebut disimpan dalam format *Extensible Markup Language* (XML). Format XML menggunakan elemen yang ditandai dengan *tag* pembuka (diawali dengan '<' dan diakhiri dengan '>'), *tag* penutup (diawali dengan '<' diakhiri '>') dan atribut elemen (parameter yang dinyatakan dalam *tag* pembuka misal <form name="isidata">). Pada data penelitian ini akan menggunakan *tag* <DOC> sebagai *root tag*, sedangkan untuk *child tag* menggunakan *tag* <DOCNO>, <TITLE>, <LABEL>, <AUTHOR> dan <TEXT>.

Praproses

Praproses merupakan proses persiapan yang dilakukan terhadap dokumen sehingga dokumen siap untuk diolah. Ada beberapa tahapan yang dilakukan didalam praproses, yaitu tokenisasi, pembakuan kata, dan pembuangan *stopwords*.

1. Tokenisasi

Tokenisasi adalah pemotongan dokumen teks menjadi bagian-bagian kecil yang disebut token dan membuang karakter-karakter tertentu seperti tanda baca dan simbol-simbol (Manning *et al.* 2009). Token bisa berupa paragraf, kalimat, frasa kata tunggal sederhana, dan konsep. *Whitespace* (spasi, tab, *newline*) digunakan sebagai pemisah antar kata yang akan dipotong. Hal ini dilakukan karena memakan waktu komputasi yang lama, sehingga proses pemotongan menjadi token dan penyimpanan ke dalam *database* disimpan diakhir setelah proses pembuangan *stopwords*.

2. Pembakuan Kata

Pembakuan kata merupakan proses penggantian kata yang tidak baku menjadi kata baku. Kata baku akan cenderung lebih kecil ambiguitas dibandingkan dengan kata yang tidak baku (Pantouw 2017). Contohnya penggunaan kata sya, ku, dan gue akan diubah menjadi kata saya, agar kata–kata tersebut memiliki makna yang sama. Pada penelitian ini proses pengantian kata

tidak baku menjadi baku menggunakan daftar kata yang sudah ada. Daftar kata yang digunakan adalah sebuah kamus yang berisi kumpulan data tidak baku dengan kata bakunya dari penelitian Aziz (2013). Hal ini dilakukan untuk memudahkan proses penggantian kata.

3. Pembuangan Stopwords

Stopwords merupakan kata–kata atau *term* yang tidak berhubungan dan tidak memiliki makna atau informasi yang berhubungan dengan dokumen, walaupun kata tersebut sering muncul pada dokumen, sehingga kata-kata tersebut memiliki nilai informasi nol (Meyer *et al.* 2008). Pembuangan kata tersebut tidak akan mengubah makna dan isi dari informasi *hoax* dan *non-hoax*. Beberapa contoh *stopwords* dalam bahasa Indonesia diantaranya: yang, juga, dari, dia, kami, kamu, aku, saya, ini, dan itu.

Seleksi Fitur

TFIDF adalah perhitungan yang menggambarkan seberapa pentingnya kata (*term*) dalam sebuah dokumen dan korpus. Proses ini digunakan untuk menilai bobot relevansi *term* dari sebuah dokumen terhadap seluruh dokumen dalam korpus. *Term Frequency* adalah ukuran seringnya kemunculan sebuah *term* dalam sebuah dokumen dan juga dalam seluruh dokumen di dalam korpus, sedangkan *inverse document frequency* adalah logaritma dari rasio jumlah seluruh dokumen dalam korpus dengan jumlah dokumen yang memiliki *term* yang dimaksud (Saadah *et al.* 2013). Menurut Manning *et al.* (2009) TF.IDF ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *Term Frequency* (TF) ke-t dan *inverse document frequency* ke-t atau idf_t dapat dirumuskan sebagai berikut:

$$idf_t = log \frac{N}{df_t} \quad (1)$$

dengan N adalah jumlah dokumen di koleksi dan df_t adalah jumlah dokumen yang mengandung term t.

Pembagian Data

K-fold cross validation adalah salah satu teknik pembagian data yang paling sering digunakan untuk mengestimasi performa atau kualitas suatu model. Dalam *k-fold cross validation* data akan dibagi ke dalam k buah partisi atau disebut dengan fold D1, D2, D3,..., Dk dengan ukuran yang sama. Pelatihan dan pengujian dilakukan sebanyak k kali seperti pada Gambar 3. Dalam iterasi ke-i, partisi Di akan menjadi data uji, selainnya menjadi data latih. Pada iterasi pertama, D1 akan menjadi data uji, D2, D3,...., Dk akan menjadi data latih. Selanjutnya iterasi kedua, D2 akan menjadi data uji, D1, D3,..., Dk akan menjadi data latih dan seterusnya (Han *et al.* 2011). Tidak seperti metode *holdout* atau *random subsampling*, disini setiap sampel memiliki waktu pelatihan yang sama dan satu kali menjadi data uji, dan untuk mengukur akurasi pada klasifikasi dengan menghitung jumlah klasifikasi yang benar untuk semua iterasi k dan dibagi dengan jumlah k partisi.

Klasifikasi

Klasifikasi Rocchio merupakan klasifikasi dengan bentuk linear, dimana klasfikasi linear menerapkan prinsip dasar *contiguity hypothesis*, bahwa dokumen dalam suatu kelas yang sama tidak akan terjadi *overlap* dengan kelas yang berbeda (Manning *et al.* 2009). Nilai *centroid* diperoleh dengan menghitung rata-rata vektor pada semua dokumen data latih untuk setiap kelasnya. *Centroid* kelas c dihitung dengan persamaan:

$$\vec{u}(c) = \frac{1}{Dc} \sum_{d \in Dc} \vec{v}(d) \quad (2)$$

dengan Dc adalah gugus dokumen di kelas c, $\vec{v}(d)$ adalah vektor kata-kata dalam kelas c, dan $\vec{u}(c)$ adalah *centroid* masing-masing kelas. Salah satu cara untuk menentukan kecocokan

dokumen uji terhadap kelas adalah dengan menghitung *cosine similarity* antara kedua titik (d_1 dan d_2) yang didefinisikan dengan persamaan:

$$sim(d_1, d_2) = \frac{\vec{v}(d_1).\vec{v}(d_2)}{\|\vec{v}(d_1)\|.\|\vec{v}(d_2)\|} \quad (3)$$

dengan titik $\vec{v}(d_1)$sebagai nilai vektor *centorid* untuk setiap kelasnya dan titik $\vec{v}(d_2)$sebagai nilai vektor data uji, $\|\vec{v}(d_1)\|$ merupakan nilai panjang vektor *centorid* untuk setiap kelasnya dan $\|\vec{v}(d_2)\|$ merupakan nilai panjang vektor data uji. Menurut Sree dan Murthy (2012), *cosine similarity* adalah teknik pengukuran kesamaan antara dua vektor dimensi *n* dengan mencari *cosinus* dari sudut antara kedua vektor tersebut. Metode *cosine similarity* ini banyak digunakan untuk menghitung kemiripan (*similarity*) antar dokumen.

Evaluasi

Tahapan evaluasi adalah tahapan untuk mengetahui ting-kat akurasi dan kinerja dari hasil klasifikasi. Kinerja klasifikasi dievaluasi dengan cara menghitung nilai *recall, precision*, dan akurasi. Menurut Manning *et al.* (2009) terdapat dua parameter yang umum digunakan untuk mengukur kinerja sebuah sistem temu kembali informasi, yaitu *precision* dan *recall*. Untuk memudahkan melakukan perhitungan, maka digunakan tabel *confusion matrix*. Menurut Manning *et al.* (2009), *confusion matrix* atau disebut juga matriks klasifikasi adalah suatu alat visual yang biasanya digunakan dalam *supervised learning*.

HASIL DAN PEMBAHASAN

Pengumpulan Dokumen

Pengumpulan dokumen dilakukan dengan mengumpulkan berita dari berbagai sumber, dimana total data berita yang terkumpul terdiri dari 163 berita *hoax* dan 300 berita *non hoax* didapatkan dari *website*, 35 berita *hoax* didapatkan dari *social media* dan 102 berita *hoax* didapatkan dari *broadcast message*. Sumber *website* didapat dari halaman viva.co.id, seword.com, pojoksatu.id, kompas.com, tempo.co, detik.com, cnn.com, republika.co.id, kumparan.com,cumicumi.com,jawapos.com, okezone.com,liputan6.com. Sedangkan *media social* didapat dari facebook dan beberapa situs blog yang menyebarkan berita dan *broadcast messages* didapat dari pesan *whatsapp, Instagram* dan *blackberry messagers.* Topik yang diambil beragam ada berupa mengenai politik, tekhnologi, islam, kesehatan, kehidupan sehari-hari dan gossip.

Praproses

Tahapan praproses dilakukan pada 600 data *hoax* dan non-*hoax* yang telah dikumpulkan dari Juli 2017 sampai dengan 15 Oktober 2017. Data tersebut kemudian dilakukan pembersihan dengan mengganti karakter "&" menjadi kata "dan" serta menghapus karakter yang tidak ada dalam daftar UTF-8. Data yang telah bersih dilakukan proses tokenisasi, pembakuan kata, dan pembuangan *stopwords*.

Pada proses tokenisasi dilakukan dengan membaca setiap kata tunggal pada sebuah dokumen, kata tunggal merupakan unit terkecil sebuah dokumen yang dipisahkan oleh karakter selain huruf. Setelah hasil tokenisasi diperoleh, proses pembakuan kata dilakukan dengan proses pengubahan kata yang tidak baku menjadi kata baku sesuai dengan daftar kata baku dan tidak baku yang berjumlah 3719 kata dari penelitian Aziz (2013) dan 75 kata penambahan langsung pada daftar, hal ini karena ada kata yang belum termasuk dalam daftar pembakuan kata. Jumlah token dari proses pembakuan kata berkurang sebanyak 130 token atau 0.77% dari jumlah token sebelumnya, hal ini dikarenakan terdapat beberapa kata yang mengalami pembakuan kata menjadi beberapa huruf yang sudah baku sehingga menjadi satu kata.

Selanjutnya, kata yang sudah menjadi baku dilakukan proses pembuangan *stopwords* sesuai daftar kata *stopwords (stoplist)* yang diperoleh dari penelitian Tala (2003) berjumlah 758

kata. Hasil pembuangan *stopwords* mengalami penurunan jumlah token dari hasil proses sebelumnya sebanyak 601 token atau sebesar 3.55% dari jumlah token sebelumnya. Data yang telah melalui tahapan pembuangan *stopwords* kemudian akan dipecah menjadi *term* membentuk matriks *term documents* atau dikenal dengan matriks TDM (*Term Document Matrix).* Pemecahan diakhir dilakukan untuk mempercepat waktu komputasi pada saat tahapan pembakuan kata dan pembuangan *stopword* di awal. Dari keseluruhan tahapan yang dilakukan menghasilkan jumlah token seperti pada Tabel 1.

Tabel 1 Jumlah token yang dihasilkan dari setiap tahapan

Tahapan	Jumlah token yang dihasilkan
Tokenisasi	17039
pembakuan kata	16909
Pembuangan *Stopwords*	16308

Dokumen yang telah melalui tahapan praproses dilakukan pengamatan karakterisitiknya untuk melihat sebaran data. Pengamatan karakteristik dilakukan dengan menghitung nilai minimum, maksimum dan rata-rata dari panjang dokumen. Perhitungan panjang dokumen dengan menghitung kuadrat dari nilai term yang muncul dalam dokumen dijumlahkan kemudian diakarkan. Hasil pengamatan terhadap panjang dokumen dapat dilihat pada Tabel 2.

Pada Tabel 2, dapat dilihat bahwa sebaran data untuk keseluruh dokumen memiliki rentang nilai 2.449 sampai 75.119 untuk kedua kelas dengan rata-rata nilai 20.188 untuk panjang dokumen kedua kelas.

Tabel 2 Deskripsi sebaran data berdasarkan panjang dokumen dari hasil praproses

Deksripsi sebaran data	Hasil perhitungan statistik
Nilai minimum	2.449
Nilai maksimum	75.119
Nilai rata-rata	20.188

K-Fold Cross Validation

Data dibagi menjadi 10 *subset* data, sehingga setiap *subset* data memiliki 60 data uji dan 540 data latih dengan pembagian 90% dan 10%. Pemilihan data uji dan data latih dilakukan secara acak di R Studio. Setiap *subset* dalam penelitian ini disebut sebagai model 1 sampai model 10. Model 1 yaitu ketika *subset* pertama dijadikan data uji dan *subset* lainnya menjadi data latih. Model 2 yaitu ketika *subset* kedua menjadi data uji dan sisanya menjadi data latih. Iterasi dilakukan sampai dengan model 10. Pembagian data uji dan data latih untuk setiap kelasnya berjumlah 270 untuk kelas *hoax* dan 270 untuk kelas *non-hoax* pada data latih, sedangkan data uji berjumlah 30 untuk kelas *hoax* dan 30 untuk kelas *non-hoax*.

Seleksi Fitur

Kata unik yang diperoleh sebanyak 15499 dari tahapan praproses kemudian diproses pada tahapan pemilihan fitur dengan menghitung nilai IDF (*Inverse Documents Frequency)* dengan mencari nilai DF (*Documents Frequency)*. Berdasarkan hokum Zipf, pemilihan fitur dilakukan dengan memberikan peringkat dengan mengurutkan kepentingan kata. Hasil perhitungan DF akan membentuk tingkat kepentingan suatu kata atau *terms*. Hukum Zipf adalah sebuah hukum yang dipopulirasikan George Kingley Zipf yang mengatakan bahwa frekuensi kata - kata yang digunakan di sebuah bahasa mengikuti sebuah pola dimana pola tersebut dijadikan parameter dalam proses selanjutnya. Pemotongan dilakukan pada kedua titik untuk mendapatkan *significant words,* yaitu kata-kata yang memiliki pengaruh signifikan pada proses klasifikasi. Pada penelitian ini, Nilai rentang batas bawah atau *lower cut-off* pada nilai DF yang digunakan adalah 400 dengan nilai IDF (*Inverse Documents Frequnecy)* 0.13033377, maka nilai DF atau kemunculan suatu kata diatas 400 dokumen akan dihapus. Sedangkan nilai retang batas atas

atau *upper cut-off* pada nilai DF yang digunakan adalah 5 dengan nilai IDF (*Inverse Documents Frequnecy)* 2.03342376, maka nilai DF atau kemunculan suatu kata dibawah 5 dokumen akan dihapus. Jumlah token yang dihasilkan setelah seleksi fitur memiliki nilai berbeda pada setiap *subset*, karena pemilihan data uji dan data latih dilakukan secara acak. Adapun jumlah kata unik yang dihasilkan sebelum seleksi fitur, setelah seleksi fitur, dan jumlah kata unik yang terpotong pada setiap percobaan seperti yang ditampilkan pada Tabel 3.

Berdasarkan tabel 3 terlihat rata-rata jumlah kata unik dari 10 *subset* adalah 15415.1 kata unik dari tahapan praproses setelah dilakukan pemilihan fitur didapatkan 2492.3 kata unik. Hal ini menunjukkan 83.8% kata unik memiliki frekuensi kemunculannya kurang dari 5 dokumen dan lebih dari 400 dokumen dan sebesar 16.2% kata unik memiliki frekuensi kemunculan lebih dari 5 dan kurang dari 400 dokumen. Nilai rentang IDF dapat mempengaruhi jumlah kata unik yang digunakan sebagai *term* dalam klasifikasi. Hal ini menunjukkan bahwa proses seleksi fitur dapat mengurangi *term* yang digunakan, karena *term* yang terpilih merupakan kata yang dianggap sebagai penciri sebuah dokumen.

Tabel 3 Jumlah kata unik dari proses seleksi fitur terhadap data latih

Model ke-	Jumlah kata unik		
	Sebelum seleksi fitur	Sesudah selesksi fitur	Jumlah kata terpotong
1	15393	2487	12906
2	15161	2467	12694
3	15423	2485	12938
4	15461	2475	12986
5	15455	2496	12959
6	15343	2498	12845
7	15465	2497	12968
8	15499	2488	13011
9	15548	2532	13016
10	15403	2498	12905

Klasifikasi

Pada tahapan klasifikasi Rocchio, kata yang didapat dari seleksi fitur dihitung nilai bobot. Nilai TF (*Term Frequency)* dikalikan dengan IDF (*Inverse Document Frequency)* untuk mendapatkan nilai bobot, dimana nilai bobot yang didapat dari data latih akan dihitung nilai *centroid* per kelas. Nilai *centroid* untuk kedua kelas disimpan dalam tabel *centroid* Rocchio yang memiliki nilai *centroid* dan *term* atau kata dari data latih. Setelah nilai *centroid* setiap kelas didapat maka dihitung jarak kemiripan dengan *cosine similarity* antara nilai *centroid* setiap kelasnya dengan nilai vektor dari data uji. Hasil *cosine similarity* dari kelas *Hoax* dan *Non-Hoax* akan dibandingkan, dimana semakin besar nilai kemiripan maka semakin besar kemiripan data uji terhadap kelas tersebut. Salah satu contoh 5 kata yang memiliki nilai *centroid* terbesar pada tabel centroid yang digunakan untuk perhitungan *cosine similarity* dapat dilihat pada Tabel 4.

Tabel 4 Lima kata yang memiliki nilai centroid tertinggi untuk model ke-5 data latih

Kelas *Hoax*	Kelas *Non-Hoax*
Islam	Korut
Indonesia	Indonesia
Ikan	Gula
Kanker	Gunung
Mie	Korea

Pada Tabel 4, contoh lima kata yang diambil dapat dilihat bahwa kata Islam, Ikan, Kanker, dan Mie sering digunakan dalam penyebaran *hoax*, sedangkan untuk berita yang melibatkan

Korut, Gula, Gunung, dan Korea jarang masyrakat menyebarkan berita palsu atau masuk dalam kelas *non-hoax.*

Klasifikasi Multinomial Naive Bayes menggunakan nilai TF (*Term Frequency)* dalam perhitungan nilai peluang frekuensi kata dari setiap kelas, dimana nilai peluang frekuensi kata tersebut disimpan dalam tabel naive bayes yang menyimpan nilai peluang frekuensi kata dalam data latih dan *term* atau kata. Data uji dilakukan pengecekan kata dalam daftar data latih, apabila kata dalam data uji tidak ada dalam daftar data latih maka akan diabaikan menurut Jurafsky dan Martin (2008). Hasil perhitungan peluang data uji terhadap data latih dilihat besarannya, semakin besar nilai peluangnya maka semakin tinggi tingkat peluang data uji masuk dalam kelas tersebut. Salah satu contoh lima kata yang memiliki nilai peluang frekuensi kata yang tinggi dalam setiap kelas pada 10 model atau *fold* dapat dilihat pada Tabel 5.

Pada Tabel 5, contoh lima kata yang diambil dari frekuensi kata terbesar terlihat bahwa negara dan anak sering digunakan untuk menyebarkan berita *hoax,* sedangkan berita yang melibatkan kata korut atau memiliki akan menunjukkan berita tersebut kelas *non-hoax.* Namun, kata indonesia, orang dan jakarta memiliki frekuensi tinggi untuk kedua kelas.

Tabel 5 Lima kata yang memiliki nilai frekuensi kata terbesar untuk model ke-5 data latih

Kelas *Hoax*	Kelas *Non-Hoax*
Indonesia	Indonesia
Orang	Orang
Negara	Jakarta
Jakarta	Memiliki
Anak	Korut

Evaluasi

Model yang digunakan adalah 10 *fold cross validation* dimana memiliki 10 model dalam pengujian klasifikasi Rocchio dan Multinomial Naive Bayes. Dari 10 model pengujian klasifikasi antara data uji dengan data latih dilakukan evaluasi dengan menghitung nilai akurasi, *precision* dan *recall*. Pada pengujian klasifikasi Rocchio akurasi terbaik didapatkan dari model 5 dengan nilai akurasi sebesar 95%, sedangkan untuk klasifikasi Multinomial Naive Bayes didapatkan dari model 5 dengan nilai akurasi masing-masing sebesar 71.67%. *Confusion matrix* dari model 5 untuk klasifikasi Rocchio dapat dilihat pada Tabel 6, sedangkan untuk model 5 pada klasifikasi Multinomial Naive Bayes dapat dilihat pada Tabel 7.

Tabel 6 *Confusion matrix* model 5 untuk klasifikasi Rocchio

Prediksi	Aktual	
	Hoax	*Non-Hoax*
Hoax	29	2
Non-Hoax	1	28

Tabel 7 *Confusion matrix* model 5 untuk klasfikasi Multinomial Naive Bayes

Prediksi	Aktual	
	Hoax	*Non-Hoax*
Hoax	28	15
Non-Hoax	2	15

Berdasarkan tabel *confusion matrix* dapat dilihat bahwa hasil klasifikasi yang salah menggunakan Rocchio berjumlah 3 dari 60 jumlah data uji, sedangkan dengan Multinomial Naive Bayes berjumlah 17 dari 60 jumlah data uji. Klasifikasi Rocchio memiliki tingkat ketepatan klasifikas yang lebih baik dibandingkan Multiomial Naive Bayes, hal ini didukung dengan jumlah kesalahan klasifikasi data uji yang lebih rendah dari Multinomial Naive Bayes. Hasil *precision* dan *recall* dari model 5 untuk klasifikasi Rocchio dan klasifikasi Multinomial Naive Bayes dapat dilihat pada Tabel 8.

Tabel 8 Nilai precision dan recall dari model 5 untuk klasifikasi Rocchio dan Multinomial Naive Bayes

Klasifikasi	*Precision*	*Recall*
Rocchio	96.67	93.55
Multinomial Naive Bayes	65.12	93.33

Akurasi rata-rata yang didapatkan dari klasifikasi Rocchio adalah 83.501% sedangkan untuk Multinomial Naive Bayes didapatkan akurasi rata-rata 65.835%. Kedua nilai akurasi tersebut memiliki selisih nilai sebesar 17.666% dan nilai akurasi untuk 10 model pada kedua klasifikasi dapat dilihat pada Tabel 9. Hasil rata-rata akurasi Multinomial Naive Bayes memiliki nilai yang lebih kecil dari Rocchio dikarenakan terjadinya kasus *underflow* atau nilai pada peluang yang dihasilkan sangat kecil, sehingga klasifikasi Rocchio memiliki nilai yang lebih baik untuk pengujian *hoax* dan non-*hoax*.

Tabel 9 Nilai akurasi pada 10 model pada kedua klasifikasi

Model ke-	Klasifikasi Rocchio	Klasifikasi Multinomial Naive Bayes
1	81.67	65.00
2	86.67	66.67
3	85.00	70.00
4	76.67	63.33
5	95.00	71.67
6	80.00	56.67
7	83.33	60.00
8	83.33	61.67
9	81.67	71.67
10	81.67	71.67

Hasil akurasi pada kedua klasifikasi di uji dengan *t-test sample* untuk melihat kedua klasifikasi tersebut memiliki nilai akurasi yang sama atau tidak. Hasil dari pengolahan R Studio didapatkan nilai probabilitis (P-value) adalah 0.000000466607, sehingga dapat disimpulkan bahwa ada perbedaan antara akurasi Rocchio dan Multinomial Naive Bayes. Dimana hasil klasifikasi Rocchio lebih baik dibandingkan Multinomial Naive Bayes. Hal ini membuktikan penelitian Joachims (1997) bahwa klasifikasi Rocchio memiliki kinerja yang lebih baik dibandingkan Multinomial Naive Bayes karena Multinomial Naive Bayes dapat terjadi kasus *underflow* atau kasus ketika nilai peluang yang diperoleh sangat kecil misalnya 0.000000000000001 yang menyebabkan klasifikasi Multinomial Naïve Bayes sulit membaca dan menggap nilai tersebut sebagai 0, sehingga dibutuhkan pencegahan khusus untuk perhitungan Multinomial Naive Bayes.

SIMPULAN

Pada penelitian ini dapat disimpulkan beberapa hal sebagai berikut, metode klasifikasi Multinomial Naive Bayes memiliki nilai akurasi dibawah nilai klasifikasi Rocchio dengan selisih nilai 17.666%. Pada 10 model dalam penelitian ini nilai akurasi mengamali penurunan paling jauh pada model 6 untuk kedua klasifikasi dimana dari model 5 ke model 6 mengalami penurunan sebesar 15 dan mengalami kenaikan paling tinggi pada model 5 untuk klasifikasi Rocchio dimana dari model 4 ke model 5 mengalami kenaikan sebesar 10. Sedangkan pada klasifikasi Multinomial Naïve Bayes mengalami kenaikan paling tinggi pada model 9 dimana model 8 ke model 9 mengalami kenaikan sebesar 10.

SARAN

Klasifikasi Rocchio pada sistem hanya menggunakan data latih dari penelitian ini tanpa ada fitur yang berfungsi untuk menambahkan model baru dari data latih. Saran untuk penelitian selanjutnya dapat menambah fitur yang berguna untuk penambahan model data latih baru yang tidak terdapat dalam system.

DAFTAR PUSTAKA

Anna, L K. 2017. *Mengapa Hoax Mudah Menyebar*. [Internet].[Diunduh tanggal 15/8/2017]. Dapat diunduh dari: http://nationalgeographic.co.id/berita/2017/ 07/mengapa-berita-hoax-mudah-menyebar.

Aziz, ATA. 2013. Sistem Pengklasifikasi Entitas pada Pesan Twitter Menggunakan Ekspresi Regular dan Naive Bayes [skripsi]. Bogor (ID): Institut Pertanian Bogor.

Chen, Y Y, Yong, S P, dan Ishak A. 2014. *Email Hoax Detection System Using Levenshtein Distance Method.* Journal of computers, Vol 9. No 2, academy publisher.

Han, J, Kamber, M, dan Pei, J. 2011. *Data Mining: Concept and Techniques, Thrid Edition*. Waltham: Morgan Kaufmann Publishers. [Internet]. [Diunduh tanggal 8/6/2017]. Dapat diunduh dari: http://myweb.sabanciuniv.edu/rdehk harghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining-Concepts-and-Tech-niques-3rd-Edition-Morgan-Kaufmann-2011.pdf.

Joachims, T. 1997. "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization" dalam: *International Conference on Machine Learning,* pp. 143-151.[Diunduh tanggal 11/8/2017]. Dapat diunduh dari: https://www.cs.cornell.edu/people/tj/publications/joachims_97a.pdf .

Manning, C, Raghavan, P, dan Schutze, H. 2009. *Introduction to Information Retrieval.* Cambridge: Cambridge University Press. [Internet]. [Diunduh tanggal 19/9/2016]. Dapat diunduh dari: https://nlp.stanford.edu/IR-book /pdf/irbookon linereading.pdf.

Meyer, D, Hornik, K, dan Feinerer, I. 2008. "Text Mining Infrastructure in R" dalam: *Journal of Statistical Software* 25(5). ISSN: 1548-7660.

Pakpahan, R. 2017. "Analisa Fenomena Hoax diberbagai Media Sosial dan Cara Menanggulangi Hoax" dalam: *Konferensi Nasional Ilmu Sosial dan Teknologi (KNIST),* pp 479-484. [Internet]. [Diunduh tanggal 3/8/2017]. Dapat diunduh dari: http://seminar.bsi.ac.id/knist/index.php/knist/article/download/474/328.

Pantouw, J C. 2017. "Perbandingan Klasifikasi Rocchio dan Multinomial Naive Bayes pada Analisis Sentimen Data Twitter Bahasa Indonesia". Skripsi. Departemen Ilmu Komputer, Institut Pertanian Bogor. 36pp.

Petkovic, T Kostanj, Z dan Pale P. 2005. *E-mail System for Automatic Hoax Recognition.* Departement of Electronic and Information Processing Faculty of Electrical Engineering and Computing, University of Zagreb.

Rasywir, E dan Purwarianti, A. 2015. "Eksperimen pada Sistem Klasifikasi Berita Hoax Berbahasa Indonesia Berbasis Pembelajaran Mesin". dalam: *Jurnal Cybermatika 3* (2). [Internet]. [Diunduh tanggal 3/8/2017]. Dapat diunduh dari: http://cybermatika.stei.itb.ac.id/ojs/index.php/cybermatika/article/download/133/65.

Saadah, M N, Atmagi, R W, Rahayu, D S, dan Arifin, A Z. 2013. "Sistem Temu Kembali Informasi Dokumen Teks dengan Pembobotan TF-IDF dan LCS" dalam: *Jurnal Teknik Informatika: Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember* 11 (1), pp. 17-20. [Internet]. [Diunduh tanggal 21/8/2017]. Dapat diunduh dari: http://download.portalgaruda.org/ article.php?article=151483&val=5910&title=SISTEM%20TEMU%20KEMBALI%20DOKUMEN%20TEKS%20DENGAN%20PEMBOBOTAN%20TF-IDF%20DAN% 20LCS.

Sree, S, Murthy, J V R. 2012. Clustering based on cosine similarity measure. *International Journal of Engineering Sciene & Advanced Technology.* 2(3) : 508-512. [Internet].

[Diunduh tanggal 03/12/2017]. Dapat diunduh dari : https://pdfs.semanticscholar.org/f009/fc0d202f4a3546cc471534714b472136dca9.pdf.

Tala, F Z. 2003. A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia. Institute for Logic, Language and Computation Universeit Van Amstredam .

Vukovic, M, Pripuzic, K, dan Belani, H. 2009. *An Intelligent Automatic Hoax Detection System.* Knowledge-Based and Intelligent Information and Engineering Systems Lecture Notes in Computer Science, Vol 5711, 318-325.